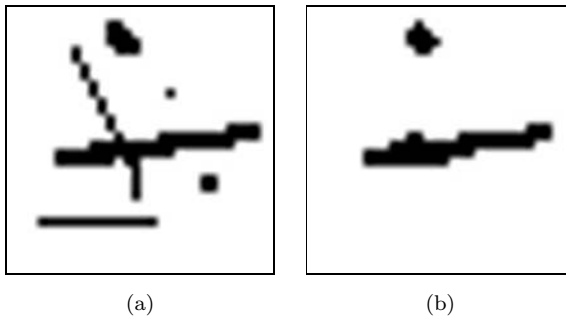


## Morphologische Filter

Bei der Diskussion des Medianfilters in Kap. 6 konnten wir sehen, dass dieser Typ von Filter in der Lage ist, zweidimensionale Bildstrukturen zu verändern (Abschn. 6.4.2). Interessant war zum Beispiel, dass Ecken abgerundet werden und kleinere Strukturen, wie einzelne Punkte und dünne Linien, infolge der Filterung überhaupt verschwinden können (Abb. 10.1). Das Filter reagiert also selektiv auf die *Form* der lokalen Bildinformation. Diese Eigenschaft könnte auch für andere Zwecke nützlich sein, wenn es gelingt, sie nicht nur zufällig, sondern kontrolliert einzusetzen. In diesem Kapitel betrachten wir so genannten „morphologische“ Filter, die imstande sind, die *Struktur* von Bildern gezielt zu beeinflussen.



**Abbildung 10.1**

$3 \times 3$  Medianfilter angewandt auf ein Binärbild. Originalbild (a) und Ergebnis nach der Filterung (b).

Morphologische Filter sind – in ihrer ursprünglichen Form – primär für Binärbilder gedacht, d. h. für Bilder mit nur zwei verschiedenen Pixelwerten 0 und 1 bzw. schwarz und weiß. Binäre Bilder finden sich an vielen Stellen, speziell im Digitaldruck, in der Dokumentenübertragung und -archivierung, aber auch als Bildmasken bei der Bildbearbeitung

und im Video-Compositing. Binärbilder können z. B. aus Grauwertbildern durch eine einfache Schwellwertoperation (Abschn. 5.1.4) erzeugt werden, wobei wir Bildelemente mit dem Wert 1 als *Vordergrund*-Pixel (*foreground*) bzw. mit dem Wert 0 als *Hintergrund*-Pixel (*background*) definieren. In den meisten der folgenden Beispiele ist, wie auch im Druck üblich, der Vordergrund schwarz dargestellt und der Hintergrund weiß.

Am Ende dieses Kapitels werden wir sehen, dass morphologische Filter nicht nur für Binärbilder anwendbar sind, sondern auch für Grauwertbilder und sogar Farbbilder, allerdings unterscheiden sich diese Filter deutlich von den binären Operationen.

## 10.1 Schrumpfen und wachsen lassen

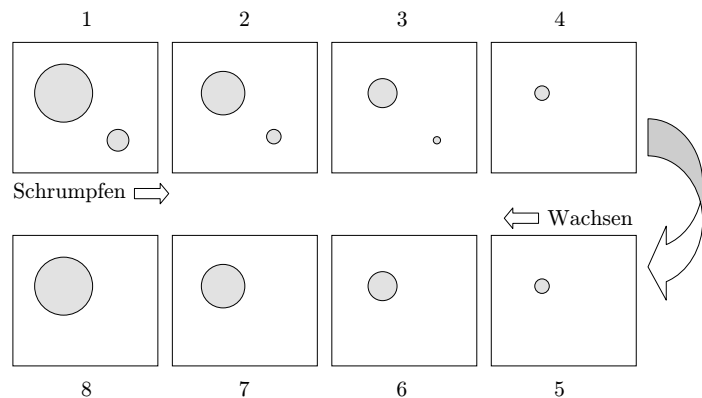
Ausgangspunkt ist also die Beobachtung, dass, wenn wir ein gewöhnliches  $3 \times 3$ -Medianfilter auf ein Binärbild anwenden, sich größere Bildstrukturen abrunden und kleinere Bildstrukturen, wie Punkte und dünne Linien, vollständig verschwinden. Das könnte nützlich sein, um etwa Strukturen unterhalb einer bestimmten Größe aus dem Bild zu eliminieren. Wie kann man aber die Größe und möglicherweise auch die Form der von einer solchen Operation betroffenen Strukturen kontrollieren?

Die strukturellen Auswirkungen eines Medianfilters sind zwar interessant, aber beginnen wir mit dieser Aufgabe nochmals von vorne. Nehmen wir also an, wir möchten kleine Strukturen in einem Binärbild entfernen, ohne die größeren Strukturen dabei wesentlich zu verändern. Dazu könnte die Kernidee folgende sein (Abb. 10.2):

1. Zunächst werden alle Strukturen im Bild schrittweise „geschrumpft“, wobei jeweils außen eine Schicht von bestimmter Stärke abgelöst wird.
2. Durch das Schrumpfen verschwinden kleinere Strukturen nach und nach, und nur die größeren Strukturen bleiben übrig.
3. Schließlich lassen wir die verbliebenen Strukturen wieder im selben Umfang wachsen.

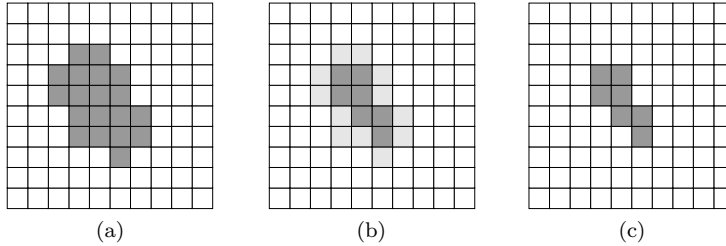
**Abbildung 10.2**

Durch schrittweises Schrumpfen und anschließendes Wachsen können kleinere Bildstrukturen entfernt werden.

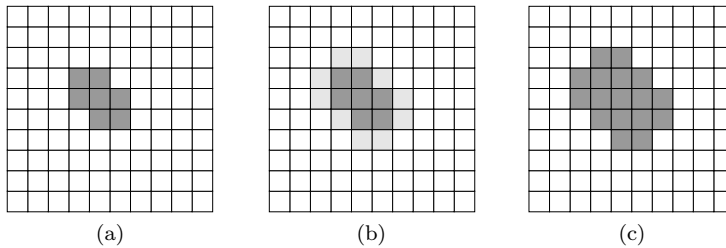


4. Am Ende haben die größeren Regionen wieder annähernd ihre ursprüngliche Form, während die kleineren Regionen des Ausgangsbilds verschwunden sind.

Alles was wir dafür benötigen, sind also zwei Operationen: „Schrumpfen“ lässt sich eine Vordergrundstruktur, indem eine Schicht außen liegender Pixel, die direkt an den Hintergrund angrenzen, entfernt wird (Abb. 10.3). Umgekehrt bedeutet das „Wachsen“ einer Region, dass eine Schicht über die direkt angrenzenden Hintergrundpixel angefügt wird (Abb. 10.4).



**Abbildung 10.3**  
Schrumpfen einer Bildregion durch Entfernen der Randpixel. Originalbild (a), markierte Randpixel, die direkt an den Hintergrund angrenzen (b), geschrumpftes Ergebnis (c).



**Abbildung 10.4**  
Wachsen einer Bildregion durch Anfügen zusätzlicher Bildelemente. Originalbild (a), markierte Hintergrundpixel, die direkt an die Region angrenzen (b), gewachsenes Ergebnis (c).

### 10.1.1 Nachbarschaft von Bildelementen

In beiden Operationen müssen wir festlegen, was es bedeutet, dass zwei Bildelemente aneinander angrenzen, d. h. „benachbart“ sind. Bei einem rechteckigen Bildraster werden traditionell zwei Definitionen von Nachbarschaft unterschieden (Abb. 10.5):

- **4er-Nachbarschaft:** die vier Pixel, die in horizontaler und vertikaler Richtung angrenzen;
- **8er-Nachbarschaft:** die Pixel der 4er-Nachbarschaft plus die vier über die Diagonalen angrenzenden Pixel.

**Abbildung 10.5**

Definition der Nachbarschaft bei rechteckigem Bildraster. 4er-Nachbarschaft  $N_1 \dots N_4$  (links) und 8er-Nachbarschaft  $N_1 \dots N_8$  (rechts).



## 10.2 Morphologische Grundoperationen

Schrumpfen und Wachsen sind die beiden grundlegenden Operationen morphologischer Filter, die man in diesem Zusammenhang als „Erosion“ bzw. „Dilation“ bezeichnet. Diese Operationen sind allerdings allgemeiner als im obigen Beispiel illustriert. Insbesondere gehen sie über das Abschälen oder Anfügen einer einzelnen Schicht von Bildelementen hinaus und erlauben wesentlich komplexere Veränderungen.

### 10.2.1 Das Strukturelement

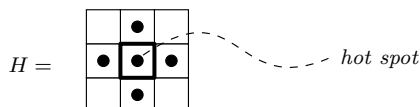
Ähnlich der Koeffizientenmatrix eines linearen Filters (Abschn. 6.2), wird auch das Verhalten von morphologischen Filtern durch eine Matrix spezifiziert, die man hier als „Strukturelement“ bezeichnet. Wie das Binärbild selbst enthält das Strukturelement nur die Werte 0 und 1, also

$$H(i, j) \in \{0, 1\},$$

und besitzt ebenfalls ein eigenes Koordinatensystem mit dem *hot spot* als Ursprung (Abb. 10.6).

**Abbildung 10.6**

Beispiel eines Strukturelements für binäre morphologische Operationen. Elemente mit dem Wert 1 sind mit  $\bullet$  markiert.



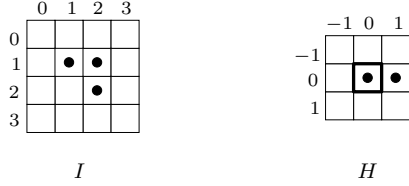
### 10.2.2 Punktmengen

Zur formalen Definition der Funktion morphologischer Filter ist es praktisch, Binärbilder als Mengen<sup>1</sup> zweidimensionaler Koordinaten-Tupel zu beschreiben. Für ein Binärbild  $I(u, v)$  besteht die zugehörige Punktmenge  $Q_I$  aus allen Koordinatenpaaren  $(u, v)$  seiner Vordergrundpixel, d. h.

<sup>1</sup> *Morphologie* (Lehre der „Gestalt“) ist u. a. ein Teilgebiet der mathematischen Mengenlehre bzw. Algebra.

$$Q_I = \{(u, v) \mid I(u, v) = 1\}. \quad (10.1)$$

Wie das Beispiel in Abb. 10.7 zeigt, kann nicht nur ein Binärbild, sondern genauso auch ein binäres Strukturelement als Punktmenge beschrieben werden.



$$Q_I = \{(1, 1), (2, 1), (2, 2)\} \quad Q_H = \{(0, 0), (1, 0)\}$$

Grundlegende Operationen auf Binärbilder können ebenfalls auf einfache Weise in dieser Mengennotation beschrieben werden. Zum Beispiel ist das Invertieren eines Binärbilds  $I(u, v) \rightarrow \neg I(u, v)$ , d. h. das Vertauschen von Vorder- und Hintergrund, äquivalent zur Bildung der Komplementärmenge

$$Q_{\neg I} = \overline{Q_I}. \quad (10.2)$$

Werden zwei Binärbilder  $I_1$  und  $I_2$  punktweise durch eine ODER-Operation verknüpft, dann ist die Punktmenge des Resultats die Vereinigung der zugehörigen Punkt Mengen  $Q_{I_1}$  und  $Q_{I_2}$ , also

$$Q_{I_1 \vee I_2} = Q_{I_1} \cup Q_{I_2}. \quad (10.3)$$

Da Punkt Mengen nur eine alternative Darstellung von binären Rasterbildern sind, werden wir beide Notationen im Folgenden je nach Bedarf synonym verwenden. Beispielsweise schreiben wir gelegentlich einfach  $I_1 \cup I_2$  statt  $Q_{I_1} \cup Q_{I_2}$ , oder auch  $\overline{I}$  statt  $\overline{Q_I}$  für ein invertiertes (Hintergrund-)Bild.

### 10.2.3 Dilation

Eine *Dilation* ist jene morphologische Operation, die unserem intuitiven Konzept des Wachsens entspricht und in der Mengennotation definiert ist als

$$I \oplus H = \{(u', v') = (u + i, v + j) \mid (u', v') \in Q_I, (i, j) \in Q_H\}. \quad (10.4)$$

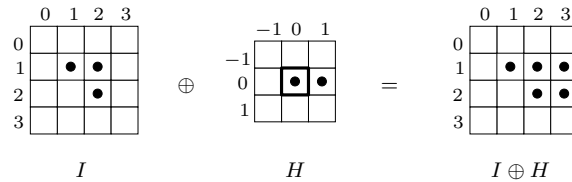
Die aus einer Dilation entstehende Punktmenge entspricht also der (Vektor-)Summe aller möglichen Kombinationen von Koordinatenpaaren aus den ursprünglichen Punkt Mengen  $Q_I$  und  $Q_H$ . Man könnte die Operation auch so interpretieren, dass das Strukturelement  $H$  an jedem Vordergrundpunkt des Bilds  $I$  repliziert wird. Oder, umgekehrt, das Bild  $I$  wird an jedem Punkt des Strukturelements  $H$  repliziert, wie das einfache Beispiel in Abb. 10.8 illustriert.

**Abbildung 10.7**

Beschreibung eines Binärbilds  $I$  und eines binären Strukturelements  $H$  als Mengen von Koordinatenpaaren  $Q_I$  bzw.  $Q_H$ .

**Abbildung 10.8**

Beispiel für Dilation. Das Binärbild  $I$  wird einer Dilation mit dem Strukturelement  $H$  unterzogen. Das Strukturelement  $H$  wird im Ergebnis  $I \oplus H$  an jedem Punkt des Originalbilds  $I$  repliziert.



$$I \oplus H = \{ (1, 1) + (0, 0), (1, 1) + (1, 0), (2, 1) + (0, 0), (2, 1) + (1, 0), (2, 2) + (0, 0), (2, 2) + (1, 0) \}$$

### 10.2.4 Erosion

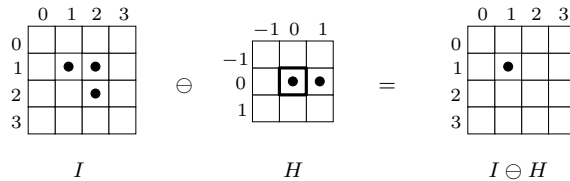
Die (quasi-)inverse Operation zur Dilation ist die *Erosion*, die wiederum in Mengennotation definiert ist als

$$I \ominus H = \{ (u', v') \mid (u' + i, v' + j) \in Q_I, \forall (i, j) \in Q_H \}. \quad (10.5)$$

Dieser Vorgang lässt sich folgendermaßen interpretieren: Ein Pixel  $(u', v')$  wird im Ergebnis dort (und nur dort) auf 1 gesetzt, wo das Strukturelement  $H$  – mit seinem *hot spot* positioniert an der Position  $(u', v')$  – vollständig im ursprünglichen Bild eingebettet ist, d. h., wo sich für jeden 1-Punkt in  $H$  auch ein entsprechender 1-Punkt in  $I$  findet. Ein einfaches Beispiel für die Erosion zeigt Abb. 10.9.

**Abbildung 10.9**

Beispiel für Erosion. Das Binärbild  $I$  wird einer Erosion mit dem Strukturelement  $H$  unterzogen. Das Strukturelement ist nur an der Position  $(1, 1)$  vollständig in das Bild  $I$  eingebettet, sodass im Ergebnis nur das Pixel mit der Koordinate  $(1, 1)$  den Wert 1 erhält.



$$I \ominus H = \{ (1, 1) \}, \text{ weil } (1, 1) + (0, 0) = (1, 1) \in Q_I \text{ und } (1, 1) + (1, 0) = (2, 1) \in Q_I$$

### 10.2.5 Eigenschaften von Dilation und Erosion

Obwohl Dilation und Erosion nicht wirklich zueinander inverse Operationen sind (i. Allg. kann man die Auswirkungen einer Erosion nicht durch eine nachfolgende Dilation nicht vollständig rückgängig machen und umgekehrt), verbindet sie dennoch eine starke formale Beziehung. Zum einen sind Dilation und Erosion *dual* insofern, als eine Dilation des Vordergrunds durch eine Erosion des Hintergrunds durchgeführt werden kann. Das Gleiche gilt auch umgekehrt, also

$$\bar{I} \oplus H = \overline{(I \ominus H)} \quad \text{und} \quad \bar{I} \ominus H = \overline{(I \oplus H)}. \quad (10.6)$$

Die *Dilation* ist des Weiteren *kommutativ*, d. h.

$$I \oplus H = H \oplus I, \quad (10.7)$$

und daher können, genauso wie bei der linearen Faltung, Bild und Strukturelement (Filter) im Prinzip miteinander vertauscht werden. Analog existiert – ähnlich der Dirac-Funktion  $\delta$  (s. Abschn. 6.3.4) – bezüglich der Dilation auch ein *neutrales Element*

$$I \oplus \delta = \delta \oplus I = I, \quad \text{wobei} \quad Q_\delta = \{(0, 0)\}. \quad (10.8)$$

Darüber hinaus ist die Dilation *assoziativ*, d. h.

$$(I_1 \oplus I_2) \oplus I_3 = I_1 \oplus (I_2 \oplus I_3), \quad (10.9)$$

also ist die Reihenfolge aufeinander folgender Dilationen nicht relevant. Das bedeutet – wie auch bei linearen Filtern (vgl. Gl. 6.21) –, dass eine Dilation mit einem großen Strukturelement der Form  $H = H_1 \oplus H_2 \oplus \dots \oplus H_K$  als Folge mehrerer Dilationen mit i. Allg. kleineren Strukturelementen realisiert werden kann:

$$I \oplus H = (\dots((I \oplus H_1) \oplus H_2) \oplus \dots \oplus H_K) \quad (10.10)$$

Die *Erosion* ist – wie die gewöhnliche arithmetische Subtraktion auch – *nicht* kommutativ, also

$$I \ominus H \neq H \ominus I. \quad (10.11)$$

Werden allerdings Erosion und Dilation miteinander kombiniert, dann gilt – wiederum analog zu Subtraktion und Addition –

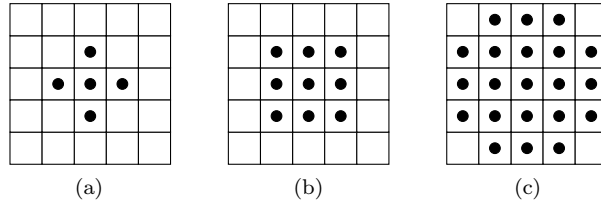
$$(I_1 \ominus I_2) \ominus I_3 = I_1 \ominus (I_2 \oplus I_3). \quad (10.12)$$

### 10.2.6 Design morphologischer Filter

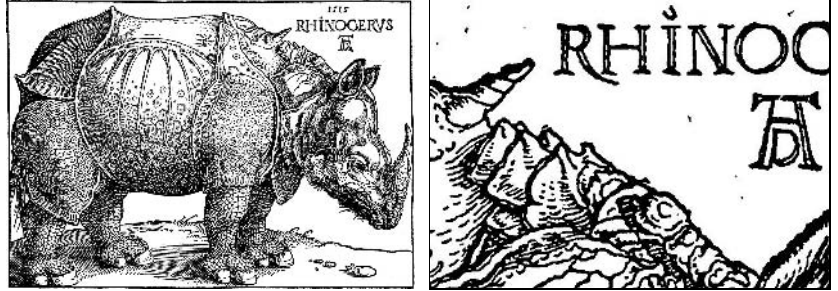
Morphologische Filter werden spezifiziert durch (a) den Typ der Filteroperation und (b) das entsprechende Strukturelement. Die Größe und Form des Strukturelements ist abhängig von der jeweiligen Anwendung, der Bildauflösung usw. In der Praxis werden häufig scheibenförmige Strukturelemente verschiedener Größe verwendet, wie in Abb. 10.10 gezeigt. Ein scheibenförmiges Strukturelement mit Radius  $r$  fügt bei einer Dilationsoperation eine Schicht mit der Dicke von  $r$  Pixel an jede Vordergrundstruktur an. Umgekehrt wird durch die Erosion mit diesem Strukturelement jeweils eine Schicht mit der gleichen Dicke abgeschält. Ausgehend von dem ursprünglichen Binärbild in Abb. 10.11, sind die Ergebnisse von Dilation und Erosion mit scheibenförmigen Strukturelementen verschiedener Radien in Abb. 10.12 gezeigt. Ergebnisse für verschiedene andere, frei gestaltete Strukturelemente sind in Abb. 10.13 dargestellt.

**Abbildung 10.10**

Typische binäre Strukturelemente verschiedener Größe. 4er-Nachbarschaft (a), 8er-Nachbarschaft (b), kleine Scheibe – *small disk* (c).

**Abbildung 10.11**

Binäres Originalbild und Ausschnitt für die nachfolgenden Beispiele (Illustration von Albrecht Dürer, 1515).



Im Unterschied zu linearen Filtern (Abschn. 6.3.3) ist es i. Allg. nicht möglich, ein *isotropes* zweidimensionales Strukturelement  $H^\circ$  aus eindimensionalen Strukturelementen  $H_x$  und  $H_y$  zu bilden, denn die Verknüpfung  $H_x \oplus H_y$  ergibt immer ein rechteckiges (also nicht isotropes) Strukturelement. Eine verbreitete Methode zur Implementierung großer scheibenförmiger Filter ist die wiederholte Anwendung kleiner scheibenförmiger Strukturelemente, wodurch sich allerdings in der Regel ebenfalls kein isotroper Gesamtoperator ergibt (Abb. 10.14).

### 10.2.7 Anwendungsbeispiel: *Outline*

Eine typische Anwendung morphologischer Operationen ist die Extraktion der Ränder von Vordergrundstrukturen. Der Vorgang ist sehr einfach: Zunächst wenden wir eine Erosion auf das Originalbild  $I$  an, um darin die Randpixel zu entfernen, d. h.

$$I' = I \ominus H_n,$$

wobei  $H_n$  ein Strukturelement z. B. für eine 4er- oder 8er-Nachbarschaft (Abb. 10.10) ist. Die eigentlichen Randpixel  $B$  sind jene, die zwar im Originalbild, aber *nicht* im erodierten Bild enthalten sind, also die Schnittmenge zwischen dem Originalbild  $I$  und dem invertierten Bild  $\bar{I}'$ ,

$$B = I \cap \bar{I}' = I \cap \overline{(I \ominus H_n)}. \quad (10.13)$$

Ein Beispiel für die Extraktion von Rändern zeigt Abb. 10.15. Interessant ist dabei, dass die Verwendung der 4er-Nachbarschaft für das Strukturelement  $H_n$  zu einer „8-verbundenen“ Kontur führt und umgekehrt [47, S. 504].





### 10.3 ZUSAMMENGESetzte OPERATIONEN

**Abbildung 10.12**  
Ergebnisse der binären Dilation  
und Erosion mit scheibenförmigen  
Strukturelementen. Der Radius  $r$  des  
Strukturelements ist 1.0 (oben), 2.5  
(Mitte) bzw. 5.0 (unten).

## 10.3 Zusammengesetzte Operationen

Aufgrund ihrer Semidualität werden Dilation und Erosion häufig in zusammengesetzten Operationen verwendet, von denen zwei so bedeutend sind, dass sie eigene Namen (und sogar Symbole) haben: „Opening“ und „Closing“.<sup>2</sup>

### 10.3.1 Opening

Ein Opening ( $I \circ H$ ) ist eine Erosion gefolgt von einer Dilation mit demselben Strukturelement  $H$ , d. h.

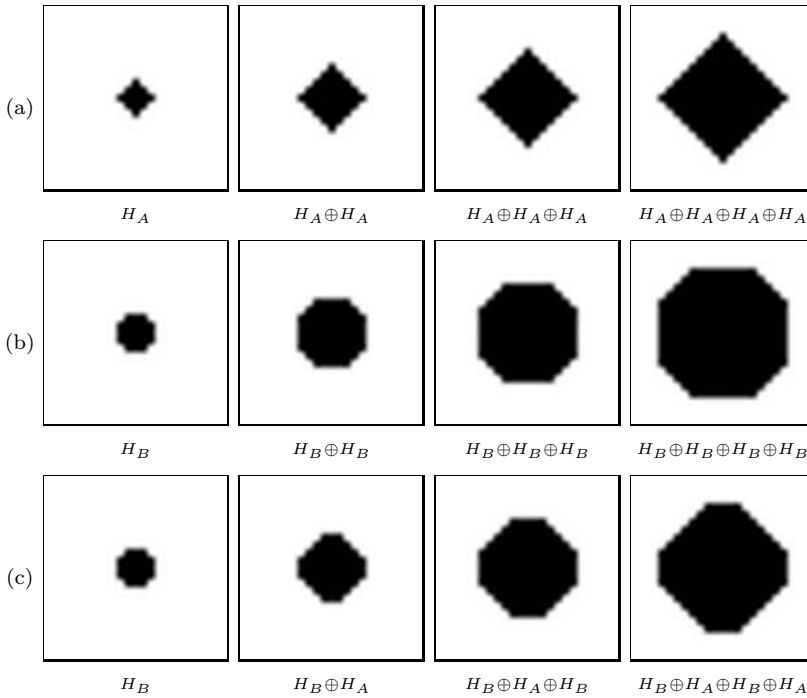
$$I \circ H = (I \ominus H) \oplus H. \quad (10.14)$$

<sup>2</sup> Die englischen Begriffe werden auch im Deutschen häufig verwendet.

**Abbildung 10.13**

Binäre Dilation und Erosion mit verschiedenen, frei gestalteten Strukturelementen. Die Strukturelemente selbst sind links gezeigt. Man sieht deutlich, dass einzelne Bildpunkte bei der Dilation zur Form des Strukturelements expandieren, ähnlich einer „Impulsantwort“. Bei der Erosion bleiben nur jene Stellen übrig, an denen das Strukturelement im Bild vollständig abgedeckt ist.

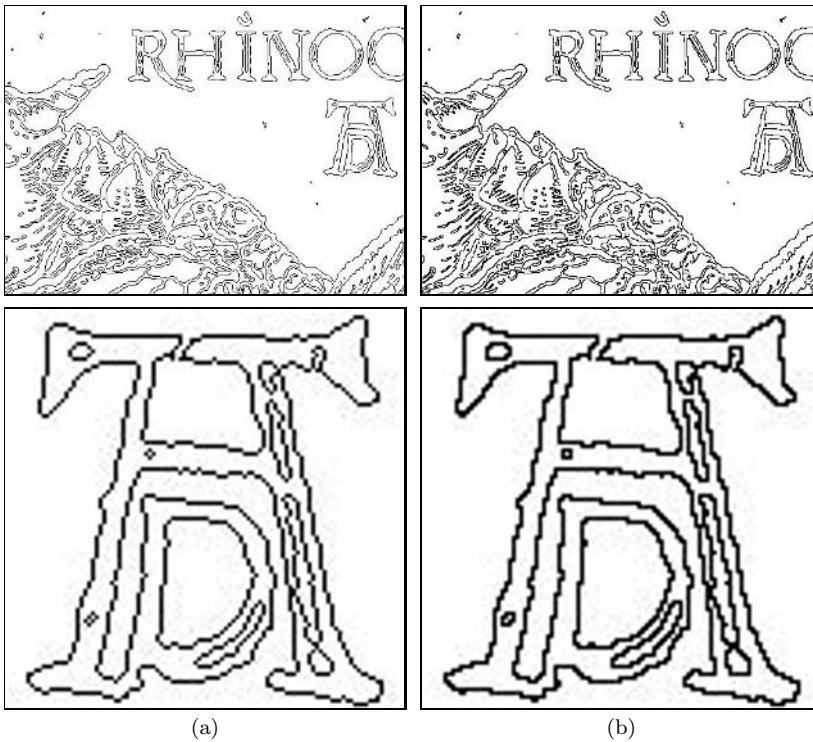




### 10.3 ZUSAMMENGESETZTE OPERATIONEN

**Abbildung 10.14**

Bildung größerer Filter durch wiederholte Anwendung kleinerer Strukturelemente. Mehrfache Anwendung des Strukturelements  $H_A$  (a) und des Strukturelements  $H_B$  (b). Abwechselnde Anwendung von  $H_B$  und  $H_A$  (c).



**Abbildung 10.15**

Extraktion von Rändern mit binären morphologischen Operationen. Das Strukturelement in (a) entspricht einer 4er-Nachbarschaft und führt zu einer „8-verbundenen“ Kontur. Umgekehrt führt ein Strukturelement mit einer 8er-Nachbarschaft (b) zu einer „4-verbundenen“ Kontur.

Ein Opening bewirkt, dass alle Vordergrundstrukturen, die kleiner sind als das Strukturelement, im ersten Teilschritt (Erosion) eliminiert werden. Die verbleibenden Strukturen werden durch die nachfolgende Dilation geglättet und wachsen ungefähr wieder auf ihre ursprüngliche Größe (Abb. 10.16). Diese Abfolge von „Schrumpfen“ und anschließend „Wachsen“ entspricht der Idee, die wir in Abschn. 10.1 skizziert hatten.

### 10.3.2 Closing

Wird die Abfolge von Erosion und Dilation umgekehrt, bezeichnet man die resultierende Operation als Closing ( $I \bullet H$ ), d. h.

$$I \bullet H = (I \oplus H) \ominus H. \quad (10.15)$$

Durch ein Closing werden Löcher in Vordergrundstrukturen und Zwischenräume, die kleiner als das Strukturelement  $H$  sind, gefüllt (Abb. 10.16).

### 10.3.3 Eigenschaften von Opening und Closing

Beide Operationen, Opening wie auch Closing, sind *idempotent*, d. h., ihre Ergebnisse sind insofern „final“, als jede weitere Anwendung derselben Operation das Bild nicht mehr verändert:

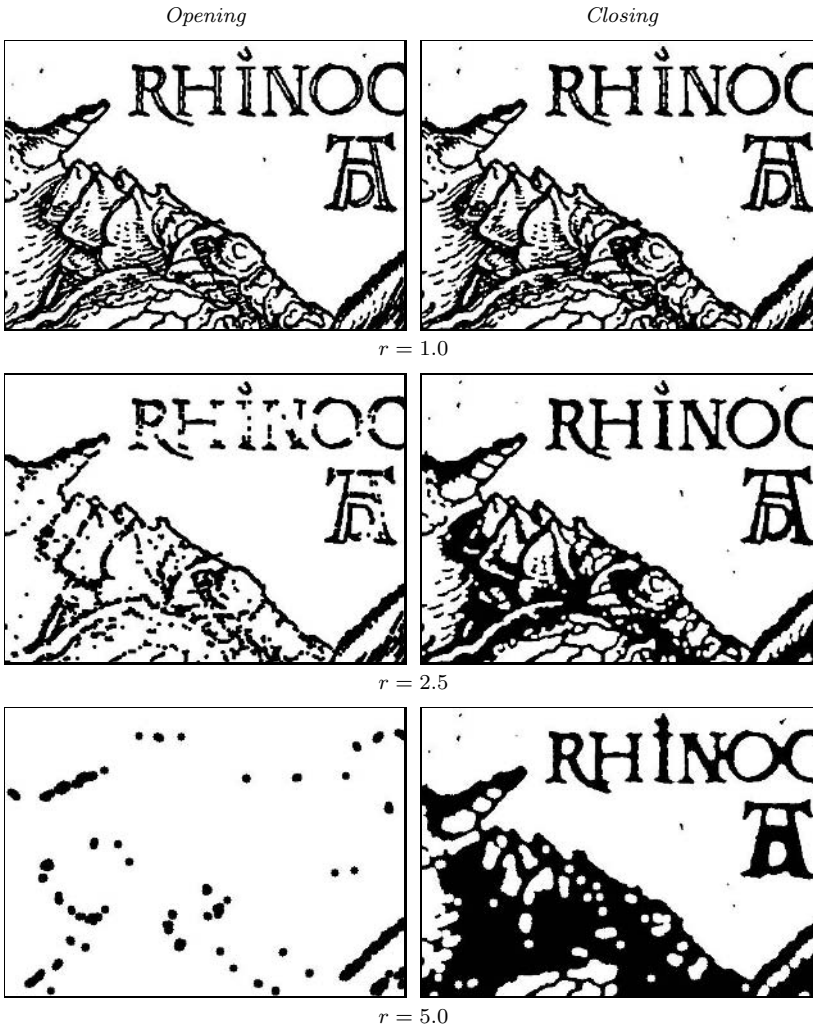
$$\begin{aligned} (I \circ H) \circ H &= I \circ H \\ (I \bullet H) \bullet H &= I \bullet H \end{aligned} \quad (10.16)$$

Die beiden Operationen sind darüber hinaus zueinander „dual“ in dem Sinn, dass ein Opening auf den Vordergrund äquivalent ist zu einem Closing des Hintergrunds und umgekehrt, d. h.

$$\begin{aligned} I \circ H &= \overline{(\overline{I} \bullet H)} \\ I \bullet H &= \overline{(\overline{I} \circ H)} \end{aligned} \quad (10.17)$$

## 10.4 Morphologische Filter für Grauwert- und Farbbilder

Morphologische Operationen sind nicht auf Binärbilder beschränkt, sondern in ähnlicher Form auch für Grauwertbilder definiert. Bei Farbbildern werden diese Methoden unabhängig auf die einzelnen Farbkanäle angewandt. Trotz der identischen Bezeichnung unterscheidet sich allerdings die Definition der morphologischen Operationen für Grauwertbilder stark von der für Binärbilder. Gleiches gilt auch für deren Anwendungen.



## 10.4 MORPHOLOGISCHE FILTER FÜR GRAUWERT- UND FARBBILDER

**Abbildung 10.16**  
Binäres Opening und Closing mit scheibenförmigen Strukturelementen. Der Radius  $r$  des Strukturelements ist 1.0 (oben), 2.5 (Mitte) bzw. 5.0 (unten).

### 10.4.1 Strukturelemente

Zunächst werden die Strukturelemente bei morphologischen Filtern für Grauwertbilder nicht wie bei binären Filtern als Punktmengen, sondern als 2D-Funktionen mit beliebigen (reellen) Werten definiert, d. h.

$$H(i, j) \in \mathbb{R}.$$

Die Werte in  $H(i, j)$  können auch negativ oder null sein, allerdings beeinflussen – im Unterschied zur linearen Faltung (Abschn. 6.3.1) – auch die Nullwerte das Ergebnis. Bei der Implementierung morphologischer Grauwert-Operationen muss daher bei den Zellen des Strukturelements  $H$  zwischen dem Wert 0 und *leeren* Zellen ( $\times$  = „don't care“) explizit unterschieden werden, also z. B.

$$\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & \mathbf{2} & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \neq \begin{array}{|c|c|c|} \hline \times & 1 & \times \\ \hline 1 & \mathbf{2} & 1 \\ \hline \times & 1 & \times \\ \hline \end{array}. \quad (10.18)$$

### 10.4.2 Grauwert-Dilation und -Erosion

Die Grauwert-Dilation  $\oplus$  wird definiert als *Maximum* der addierten Werte des Filters  $H$  und der entsprechenden Bildregion  $I$ , d. h.

$$(I \oplus H)(u, v) = \max_{(i,j) \in H} \{I(u+i, v+j) + H(i, j)\}. \quad (10.19)$$

Umgekehrt entspricht die Grauwert-Erosion dem *Minimum* der Differenzen, also

$$(I \ominus H)(u, v) = \min_{(i,j) \in H} \{I(u+i, v+j) - H(i, j)\}. \quad (10.20)$$

Abb. 10.17 zeigt anhand eines einfachen Beispiels die Wirkungsweise der Grauwert-Dilation, Abb. 10.18 demonstriert analog dazu die Erosion. In beiden Operationen können grundsätzlich negative Ergebniswerte entstehen, die bei einem eingeschränkten Wertebereich z. B. mittels *Clamping*

**Abbildung 10.17**

Grauwert-Dilation  $I \oplus H$ . Das  $3 \times 3$ -Strukturelement ist dem Bild  $I$  überlagert dargestellt. Die Werte in  $I$  werden elementweise zu den entsprechenden Werten in  $H$  addiert; das Zwischenergebnis ( $I + H$ ) für die gezeigte Filterposition ist darunter abgebildet. Dessen Maximalwert  $8 = 7 + 1$  wird an der aktuellen Position des *hot spot* in das Ergebnisbild ( $I \oplus H$ ) eingesetzt. Zusätzlich sind die Ergebnisse für drei weitere Filterpositionen gezeigt.

$$\begin{array}{|c|c|c|c|} \hline I & & H & \\ \hline 6 & 7 & 3 & 4 \\ \hline 5 & 6 & 6 & 8 \\ \hline 6 & 4 & 5 & 2 \\ \hline 6 & 4 & 2 & 3 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline I \oplus H & & & \\ \hline & & & \\ \hline & 8 & 9 & \\ \hline & 7 & 9 & \\ \hline & & & \\ \hline \end{array}$$
  

$$I + H = \begin{array}{|c|c|c|c|} \hline 7 & 8 & 4 & \\ \hline 6 & 8 & 7 & \\ \hline 7 & 5 & 6 & \\ \hline \end{array} \quad \text{max}$$

**Abbildung 10.18**

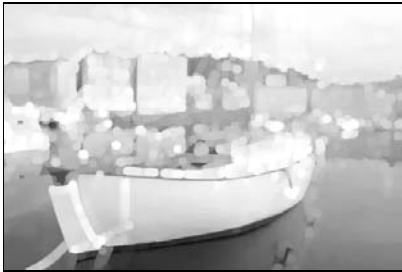
Grauwert-Erosion  $I \ominus H$ . Das  $3 \times 3$ -Strukturelement ist dem Bild  $I$  überlagert dargestellt. Die Werte von  $H$  werden elementweise von den entsprechenden Werten in  $I$  subtrahiert; das Zwischenergebnis ( $I - H$ ) für die gezeigte Filterposition ist darunter abgebildet. Dessen Minimalwert  $3 - 1 = 2$  wird an der aktuellen Position des *hot spot* in das Ergebnisbild ( $I \ominus H$ ) eingesetzt.

$$\begin{array}{|c|c|c|c|} \hline I & & H & \\ \hline 6 & 7 & 3 & 4 \\ \hline 5 & 6 & 6 & 8 \\ \hline 6 & 4 & 5 & 2 \\ \hline 6 & 4 & 2 & 3 \\ \hline \end{array} \ominus \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline I \ominus H & & & \\ \hline & & & \\ \hline & 2 & 1 & \\ \hline & 1 & 1 & \\ \hline & & & \\ \hline \end{array}$$
  

$$I - H = \begin{array}{|c|c|c|c|} \hline 5 & 6 & 2 & \\ \hline 4 & 4 & 5 & \\ \hline 5 & 3 & 4 & \\ \hline \end{array} \quad \text{min}$$

Dilation

Erosion

 $r = 2.5$  $r = 5.0$  $r = 10.0$ 

(s. Abschn. 5.1.2) zu berücksichtigen sind. Ergebnisse von Dilation und Erosion mit konkreten Grauwertbildern und scheibenförmigen Strukturelementen verschiedener Größe zeigt Abb. 10.19.

### 10.4.3 Grauwert-Opening und -Closing

Opening und Closing für Grauwertbilder sind – genauso wie für Binärbilder (Gl. 10.14, 10.15) – als zusammengesetzte Dilation und Erosion mit jeweils demselben Strukturelement definiert. Abb. 10.20 zeigt Beispiele für diese Operationen, wiederum unter Verwendung scheibenförmiger Strukturelemente verschiedener Größe.

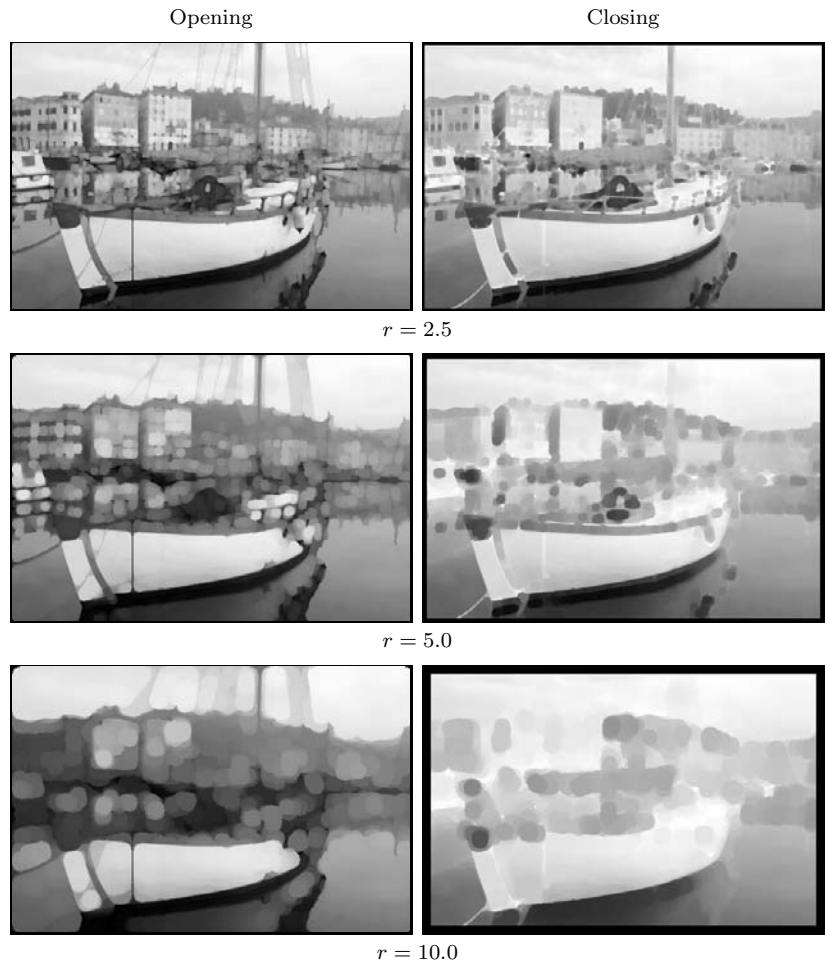
In Abb. 10.21 und 10.22 sind die Ergebnisse von Grauwert-Dilation und -Erosion bzw. von Grauwert-Opening und -Closing mit verschiedenen, frei gestalteten Strukturelementen dargestellt. Interessante Effekte können z. B. mit Strukturelementen erzielt werden, die der Form natürlicher Pinselstriche ähnlich sind.

## 10.4 MORPHOLOGISCHE FILTER FÜR GRAUWERT- UND FARBBILDER

### Abbildung 10.19

Grauwert-Dilation und -Erosion mit scheibenförmigen Strukturelementen. Der Radius  $r$  des Strukturelements ist 2.5 (oben), 5.0 (Mitte) und 10.0 (unten).

**Abbildung 10.20**  
Grauwert-Opening und -Closing  
mit scheibenförmigen Struktu-  
relementen. Der Radius  $r$  des  
Strukturelements ist 2.5 (oben),  
5.0 (Mitte) und 10.0 (unten).



## 10.5 Implementierung morphologischer Filter

### 10.5.1 Binäre Bilder in ImageJ

Binäre Bilder werden in ImageJ – genauso wie Grauwertbilder – mit 8 Bits pro Pixel dargestellt.<sup>3</sup> Üblicherweise verwendet man die Intensitätswerte 0 und 255 für die binären Werte 0 bzw. 1. In diesem Fall werden Vordergrundpixel weiß und Hintergrundpixel schwarz dargestellt. Falls eine umgekehrte Darstellung (Vordergrund schwarz) gewünscht ist, kann dies am einfachsten durch Invertieren der Display-Funktion (Lookup-Table LUT) erreicht werden: entweder über das Menü

Image→Lookup Tables→Invert LUT

oder innerhalb des Programms durch die `ImageProcessor`-Methode

<sup>3</sup> In ImageJ gibt es kein spezielles Speicherformat für binäre Bilder, auch die Klasse `BinaryProcessor` verwendet 8-Bit Bilddaten.





## 10.5 IMPLEMENTIERUNG MORPHOLOGISCHER FILTER

### Abbildung 10.21

Grauwert-Dilation und -Erosion mit verschiedenen, frei gestalteten Strukturelementen.

```
void invertLut().
```

Diese Anweisung verändert nur die Anzeige des jeweiligen Bilds und nicht die Bildinhalte (Pixelwerte) selbst.

### 10.5.2 Dilation und Erosion

Die wichtigsten morphologischen Operationen sind in ImageJ als Methoden der Klasse `ImageProcessor` (s. auch Abschn. 10.5.5) bereits fertig implementiert, allerdings beschränkt auf Strukturelemente der Größe  $3 \times 3$ .

**Abbildung 10.22**  
Grauwert-Opening und -Closing  
mit verschiedenen, frei ge-  
stalteten Strukturelementen.



Im Folgenden ist als Beispiel die Implementierung der binären Dilation gezeigt, über die – wegen der Dualität zur Erosion (Gl. 10.6) – auch die meisten anderen morphologischen Operationen realisiert werden können. Ausgangspunkt für die `dilate()`-Methode ist ein Binärbild  $I$  mit den Werten 0 (Hintergrund) und 255 (Vordergrund) sowie ein zweidimensionales Strukturelement  $H$  mit 0/1-Werten, dessen *hot spot* im Zentrum angenommen wird:

```
1  import ij.process.Blitter;
2  import ij.process.ImageProcessor;
3  ...
4  void dilate(ImageProcessor I, int[] [] H){
5      //assume that the hot spot of H is at its center (ic,jc):
6      int ic = (H[0].length-1)/2;
7      int jc = (H.length-1)/2;
8
9      ImageProcessor np
10         = I.createProcessor(I.getWidth(),I.getHeight());
11
12     for (int j=0; j<H.length; j++){
13         for (int i=0; i<H[j].length; i++){
14             if (H[j][i] == 1) { // this pixel is set
15                 //copy image into position (i-ic,j-jc):
16                 np.copyBits(I,i-ic,j-jc,Blitter.MAX);
17             }
18         }
19     }
20     //copy result back to original image
21     I.copyBits(np,0,0,Blitter.COPY);
22 }
```

Für das Ergebnis wird zunächst (in Zeile 9) ein neues (leeres) Bild angelegt, das am Ende in das ursprüngliche Bild zurückkopiert wird (Zeile 21). Die eigentliche Dilation erfolgt iterativ, indem für jede Position  $(i, j)$  im Strukturelement mit dem Wert  $H(i, j) = 1$  das ursprüngliche Bild, verschoben um  $(i, j)$ , in das Ergebnis kopiert wird. Dazu wird (in Zeile 16) die ImageProcessor-Methode `copyBits()` mit dem Operationsschlüssel `Blitter.MAX` verwendet (s. auch Abschn. 5.7.3). Interpretiert man die Pixel als binäre Werte, entspricht dies einer ODER-Verknüpfung zwischen dem Ergebnis und dem verschobenen Ausgangsbild.

Die Dilation ist die einzige Operation, die tatsächlich im Detail implementiert werden muss, denn die Erosion kann als Dilation des Hintergrunds durchgeführt werden, also durch Invertieren des Bilds, Durchführung der Dilation und neuerlichem Invertieren:

```
23 void erode(ImageProcessor I, int[] [] H){
24     I.invert();
25     dilate(I,H); //dilates the background
26     I.invert();
27 }
```

### 10.5.3 Opening und Closing

Opening- und Closing-Operation können nunmehr einfach als Abfolge von Dilation und Erosion mit jeweils demselben Strukturelement  $H$  (Abschn. 10.3) realisiert werden:

```

28 void open(ImageProcessor I, int[] [] H){
29     erode(I,H);
30     dilate(I,H);
31 }
32
33 void close(ImageProcessor I, int[] [] H){
34     dilate(I,H);
35     erode(I,H);
36 }

```

#### 10.5.4 Outline

Für die in Abschn. 10.2.7 beschriebene *Outline*-Operation zur Extraktion von Rändern können wir z. B. ein  $3 \times 3$ -Strukturelement  $H$  für die 4er-Nachbarschaft verwenden. Zunächst wird eine Kopie des Bilds ( $I_e$ ) erzeugt und anschließend erodiert (Zeile 43). Zwischen dem Ergebnis der Erosion und dem Originalbild wird in Zeile 44 die Differenz gebildet (durch Anwendung der Methode `copyBits()` mit dem Argument `Blitter.DIFFERENCE`). Am Ende enthält das ursprüngliche Bild  $I$  die Umrisse der Vordergrundstrukturen:

```

37 void outline(ImageProcessor I){
38     int[] [] H = { //4-neighborhood structuring element
39         {0,1,0},
40         {1,1,1},
41         {0,1,0}};
42     ImageProcessor Ie = I.duplicate();
43     erode(Ie,H); // /*  $I' \leftarrow I \ominus H$  */
44     I.copyBits(Ie,0,0,Blitter.DIFFERENCE); //  $I \leftarrow I \cap I'$ 
45 }

```

#### 10.5.5 Morphologische Operationen in ImageJ

##### ImageProcessor

ImageJ stellt für die Klasse `ImageProcessor` einige fertige Methoden für einfache morphologische Filter zur Verfügung:

```

void dilate()
void erode()
void open()
void close()

```

Diese Methoden verwenden vollbesetzte  $3 \times 3$ -Strukturelemente (entsprechend Abb. 10.10 (b)), die – abhängig vom Bildinhalt – entweder binäre oder Grauwert-Operationen durchführen. Die Klasse `ColorProcessor` stellt diese Methoden auch für RGB-Farbbilder zur Verfügung, wobei die morphologischen Operationen individuell auf die einzelnen Farbkanäle wie für gewöhnliche Grauwertbilder angewandt werden.

Daneben bietet die Klasse `BinaryProcessor` (eine Unterklasse von `ByteProcessor`) mit den Methoden

```
void outline()
void skeletonize()
```

spezielle morphologische Operationen, die ausschließlich für Binärbilder definiert sind. Die Methode `outline()` implementiert die Extraktion von Rändern mit dem Strukturelement einer 8er-Nachbarschaft, wie in Abschn. 10.2.7 beschrieben.

Die in der Methode `skeletonize()` implementierte Operation bezeichnet man als „Thinning“, das ist eine iterative Erosion, mit der Strukturen auf eine Dicke von 1 Pixel reduziert werden, ohne sie dabei in mehrere Teile zu zerlegen. Dabei muss, abhängig vom aktuellen Bildinhalt innerhalb der Filterregion (üblicherweise von der Größe  $3 \times 3$ ), jeweils entschieden werden, ob tatsächlich eine Erosion durchgeführt werden soll oder nicht. Die Operation erfolgt in mehreren Durchläufen so lange, bis sich im Ergebnis keine Änderungen mehr ergeben (s. beispielsweise [30, S. 535], [48, S. 517]). Die konkrete Implementierung in ImageJ basiert auf einem effizienten Algorithmus von Zhang und Suen [91]. Abbildung 10.23 zeigt ein Beispiel für die Anwendung von `skeletonize()`.

Die Verwendung der Methoden `outline()` und `skeletonize()` setzt ein Objekt der Klasse `BinaryProcessor` voraus, das wiederum nur aus einem bestehenden `ByteProcessor` erzeugt werden kann. Dabei wird angenommen, dass das ursprüngliche Bild nur Werte mit 0 (Hintergrund) und 255 (Vordergrund) enthält. Das nachfolgende Beispiel zeigt den Einsatz von `outline()` innerhalb der `run()`-Methode eines ImageJ-Plugins:

```
1 import ij.process.*;
2 ...
3
4 public void run(ImageProcessor ip) {
5     BinaryProcessor bp
6         = new BinaryProcessor((ByteProcessor)ip);
7     bp.outline();
8 }
```

Der neue `BinaryProcessor bp` legt übrigens keine eigenen Bilddaten an, sondern verweist lediglich auf die Bilddaten des ursprünglichen Bilds `ip`, sodass jede Veränderung von `bp` (z. B. durch den Aufruf von `outline()`) gleichzeitig auch `ip` betrifft.

## Weitere morphologische Filter

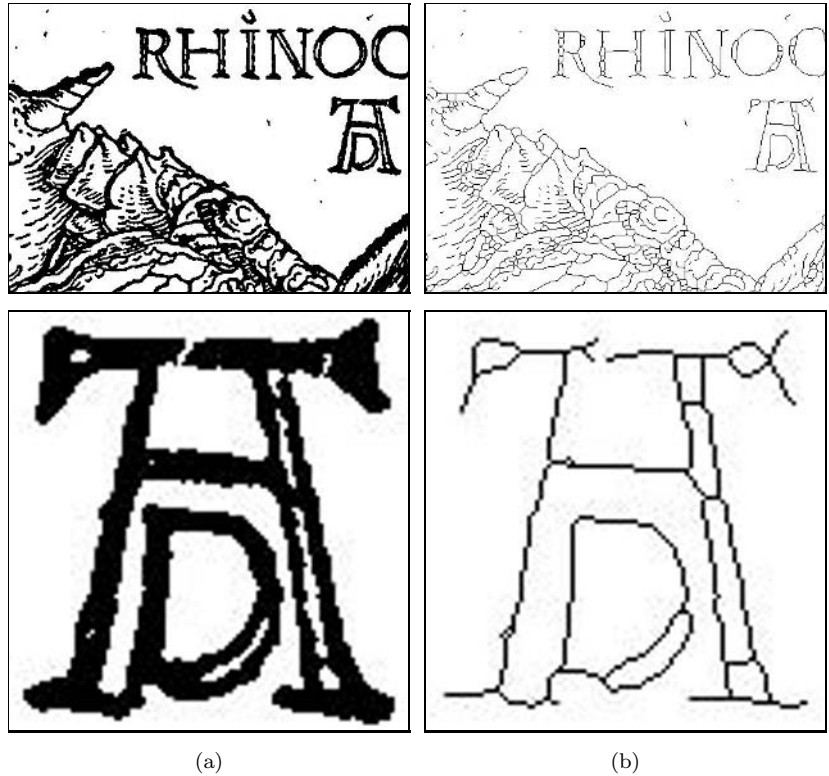
Neben den in ImageJ direkt implementierten Methoden sind einzelne Plugins und ganze Packages für spezielle morphologische Filter online<sup>4</sup>

---

<sup>4</sup> <http://rsb.info.nih.gov/ij/plugins>

**Abbildung 10.23**

Beispiel für die Anwendung  
der `skeletonize()`-Methode.  
Originalbild bzw. Detail (a)  
und zugehörige Ergebnisse (b).



verfügbar, wie z. B. das *Grayscale Morphology Package* von Dimitar Prodanov, bei dem die Strukturelemente weitgehend frei spezifiziert werden können (eine modifizierte Version wurde für einige der Beispiele in diesem Kapitel verwendet).

## 10.6 Aufgaben

**Aufg. 10.1.** Berechnen Sie manuell die Ergebnisse für die Dilation und die Erosion zwischen dem folgenden Binärbild  $I$  und den Strukturelementen  $H_1$  und  $H_2$ :

$$I = \begin{array}{|c|c|c|c|c|c|} \hline & & & & \bullet & \\ \hline & \bullet & \bullet & \bullet & \bullet & \bullet \\ \hline \bullet & & & \bullet & \bullet & \bullet \\ \hline \bullet & \bullet & & \bullet & \bullet & \\ \hline & & \bullet & & \bullet & \\ \hline & & & \bullet & & \\ \hline \end{array} \quad H_1 = \begin{array}{|c|c|c|} \hline \bullet & & \\ \hline & \bullet & \\ \hline & & \bullet \\ \hline \end{array} \quad H_2 = \begin{array}{|c|c|c|} \hline & \bullet & \\ \hline \bullet & \bullet & \bullet \\ \hline & \bullet & \\ \hline \end{array}$$

**Aufg. 10.2.** Angenommen, in einem Binärbild  $I$  sind störende Vordergrundflecken mit einem Durchmesser von maximal 5 Pixel zu entfernen

und die restlichen Bildkomponenten sollen möglichst unverändert bleiben. Entwerfen Sie für diesen Zweck eine morphologische Operation und erproben Sie diese an geeigneten Testbildern.

**Aufg. 10.3.** Zeigen Sie, dass im Fall eines Strukturelements der Form

•	•	•
•	•	•
•	•	•

für Binärbilder bzw.

0	0	0
0	<b>0</b>	0
0	0	0

für Grauwertbilder

eine Dilation äquivalent zu einem  $3 \times 3$ -Maximum-Filter und die entsprechende Erosion äquivalent zu einem  $3 \times 3$ -Minimum-Filter ist (Abschn. 6.4.1).